WHAT IS CLAIMED IS:

1. A method in a signal processor for filtering samples in a digital signal, the method comprising:

generating an approximate filtered sample as a function of less than four samples of the digital signal;

generating a correction as a function of the less than four samples; and

generating a filtered sample by modifying the approximate filtered sample with the correction.

2. The method of claim 1, wherein the less than four samples are only two samples.

3. The method of claim 2, wherein the two samples are a first fixed-point number, A, and a second fixed-point number, B, wherein generating the approximate filtered sample includes calculating (A+B+1)>>1.

4. The method of claim 3, wherein the signal processor is a microprocessor having an instruction for calculating the function (X+Y+1)>>1, and wherein calculating A+B+1>>1 is performed using the instruction.

5. The method of claim 4, wherein the microprocessor is an Intel™ microprocessor with SSE or SSE2, and wherein the instruction is the PAVGB instruction.

6. The method of claim 4, wherein the microprocessor is an Intel™ microprocessor with SSE or SSE2, and wherein the instruction is the PAVGW instruction.

7. The method of claim 3, wherein generating the approximate filtered sample further includes calculating (A + ((A+B+1)>>1) + 1)>>1.

8. The method of claim 3, wherein generating the approximate filtered sample further includes:

calculating E = ((A+B+1)>>1)<<S;

calculating F = ((A+B+1)>>1)<<R; and

calculating the approximate filtered sample as E + F;

wherein S and R are positive fixed-point numbers.

9. The method of claim 8, wherein generating the correction includes:

calculating $Q = \sim(A \oplus B)$;

masking Q with the number one;

calculating $G = Q \ll (S-1)$;

calculating $H = Q \ll (R-1)$; and

calculating the correction as $G + H$.

10. The method of claim 9, wherein generating the filtered sample includes:

calculating the filtered sample as the approximate filtered sample added with the correction; and

right-shifting the filtered sample by N-1 bits, wherein N is a positive fixed-point number.

11. The method of claim 3, wherein generating the approximate filtered sample further includes:

calculating $E = ((A+B+1) \gg 1) \gg (N-1-S)$;

calculating $F = ((A+B+1) \gg 1) \gg (N-1-R)$; and

adding E with F;

wherein N, S and R are positive fixed-point numbers, and wherein $N \geq S > R$.

12. The method of claim 11, wherein generating the correction includes:

calculating $Q = \sim(A \oplus B)$;

masking Q with the number one;

calculating $G = Q \gg (N-S)$;

calculating $H = Q \gg (N-R)$; and

calculating the correction as $G + H$.

13. The method of claim 12, wherein generating the filtered sample includes calculating the filtered sample as the approximate filtered sample added with the correction.

14. The method of claim 2, wherein the two samples are fixed-point numbers, and wherein generating the correction includes:

calculating the correction as the exclusive OR (XOR) of the two samples; and

masking the correction with the integer one.

15. The method of claim 14, wherein generating the correction further includes, prior to the masking step, generating a bit-wise complement of the correction.

16. The method of claim 2, wherein the two samples are fixed-point numbers A, B, and wherein generating the correction includes:

calculating the correction as $(A \oplus B)$ OR $(A \oplus (A+B>>1))$; and

masking the correction with the number one.

17. The method of claim 2, wherein the two samples are fixed-point numbers A, B, and wherein generating the correction includes:

calculating the correction as $(A \oplus (A+B>>1))$; and

bit-wise ANDing the correction with the number one.

18. The method of claim 1, wherein generating the filtered sample includes adding the correction to the approximate filtered sample.

19. The method of claim 1, wherein generating the filtered sample includes subtracting the correction from the approximate interpolated sample.

20. The method of claim 2, wherein the microprocessor is an Intel™ microprocessor with MMXTM/SSE, wherein the two samples are 8-bit fixed-point numbers, wherein the steps of generating the approximate filtered sample, generating the correction and generating the filtered sample include executing the instructions:

```
pxor     C_REG, A_REG ;
pand     C_REG, CONST ;
pavgb    A_REG, B_REG ;
psubb    A_REG, C_REG ;
```

wherein A_REG is a register that initially includes one of the two samples, B_REG is a register that includes the other of the two samples, C_REG is a register that initially includes the other of the two samples, and CONST is a constant that includes the eight-bit number 0x01.

21. The method of claim 2, wherein the microprocessor is an Intel™ microprocessor with MMXTM/SSE, wherein the two samples are eight bit integers, wherein

the steps of generating the approximate filtered sample, generating the correction and generating the filtered sample include executing the instructions:

```
pxor     C_REG, A_REG ;
pandn    C_REG, CONST ;
pavgb    A_REG, B_REG ;
paddb    A_REG, C_REG ;
```

wherein A_REG is a register that initially includes one of the two samples, B_REG is a register that includes the other of the two samples, C_REG is a register that initially includes the other of the two samples, and CONST is a constant that includes the eight-bit number 0x01.

22. The method of claim 1, wherein the less than four samples are only three samples.

23. The method of claim 22, wherein the two samples are a first fixed-point number, A, a second fixed-point number, B, and a third fixed-point number, C, wherein generating the approximate filtered sample includes:

calculating $M = (A+B+1)>>1$;

calculating $N = (C+0+1)>>1$; and

calculating the approximate filtered sample as $X = (M+N+1)>>1$.

24. The method of claim 23, wherein generating the correction includes:

calculating the correction as $((A\oplus B)AND(C))OR(M\oplus N)$; and

masking the correction with the number one.

25. The method of claim 23, wherein generating the correction includes:

calculating the correction as $(A\oplus B)OR\ C\ OR(M\oplus N)$; and

masking the correction with the number one.

26. The method of claim 22, wherein the two samples are a first fixed-point number, A, a second fixed-point number, B, and a third fixed-point number, C, wherein generating the approximate filtered sample includes:

calculating $M = (B+C+1)>>1$; and

calculating the approximate filtered sample as $X = (A+M+1)>>1$.

27. The method of claim 26, wherein generating the correction includes:

calculating the correction as $A\oplus M$; and

masking the correction with the number one.

28. The method of claim 26, wherein generating the correction includes:
calculating the correction as (B⊕C) OR (A⊕M); and
masking the correction with the number one.

29. A method in a signal processor for filtering samples in a digital signal, the method comprising:
generating an approximate filtered sample as a function of a first sample, A, a second sample, B, a third sample, C, and a fourth sample, D, wherein A, B, C, and D are fixed-point numbers;
generating a correction as ((A⊕B)AND(C⊕D))OR(M⊕N), wherein M=A+B+1>>1 and N=C+D+1>>1;
masking the correction; and
generating a filtered sample by modifying the approximate filtered sample with the correction.

30. The method of claim 29, wherein generating the approximate filtered sample includes:
calculating M = A+B+1>>1;
calculating N = C+D+1>>1; and
calculating the approximate filtered sample as M+N+1>>1.

31. The method of claim 30, wherein the signal processor is a microprocessor having an instruction for calculating the function X+Y+1>>1, wherein the step of calculating M is performed using the instruction, wherein the step of calculating N is performed using the instruction, and wherein the step of calculating the approximate filtered sample is performed using the instruction.

32. The method of claim 31, wherein the microprocessor is an Intel™ microprocessor with SSE or SSE2, and wherein the instruction is the PAVGB instruction.

33. The method of claim 31, wherein the microprocessor is an Intel™ microprocessor with SSE or SSE2, and wherein the instruction is the PAVGW instruction.

34. The method of claim 29, wherein the microprocessor is an Intel™ microprocessor with MMXTM/SSE, wherein the A, B, C, and D are eight bit fixed-point numbers, wherein the steps of generating the approximate filtered sample, generating the correction, masking the correction and generating the filtered sample include executing the instructions:

```
pxor    A_REG, B_REG ;
pxor    C_REG, D_REG ;
pand    A_REG, C_REG ;
pavgb   M_REG, B_REG ;
pavgb   N_REG, D_REG ;
movq    B_REG, M_REG ;
pxor    B_REG, N_REG ;
pand    B_REG, CONST ;
pavgb   M_REG, N_REG ;
psubb   M_REG, B_REG ;
```

wherein A_REG is a register that initially includes A, B_REG is a register that initially includes B, C_REG is a register that initially includes C, D_REG is a register that includes D, CONST is a constant that includes the eight-bit number 0x01, M_REG is a register that initially includes A, and N_REG is a register that initially includes C.

35. A method in a signal processor for filtering samples in a digital signal, the method comprising:

generating an approximate filtered sample as a function of a first sample, A, a second sample, B, a third sample, C, and a fourth sample, D, wherein A, B, C, and D are integers;

generating a correction as $(A \oplus B)\mathrm{OR}(C \oplus D)\mathrm{OR}(M \oplus N)$, wherein $M=A+B+1>>1$ and $N=C+D+1>>1$;

masking the correction; and

generating a filtered sample by modifying the approximate filtered sample with the correction.

36. The method of claim 35, wherein generating the approximate filtered sample includes:

calculating $M = A+B+1>>1$;

calculating $N = C+D+1>>1$; and

calculating the approximate interpolated sample as $M+N+1>>1$.

37. The method of claim 36, wherein the signal processor is a microprocessor having an instruction for calculating the function X+Y+1>>1, wherein the step of calculating M is performed using the instruction, wherein the step of calculating N is performed using the instruction, and wherein the step of calculating the approximate filtered sample is performed using the instruction.

38. The method of claim 37, wherein the microprocessor is an Intel™ microprocessor with SSE or SSE2, and wherein the instruction is the PAVGB instruction.

39. The method of claim 37, wherein the microprocessor is an Intel™ microprocessor with SSE or SSE2, and wherein the instruction is the PAVGW instruction.

40. The method of claim 35, wherein the microprocessor is an Intel™ microprocessor with MMXTM/SSE, wherein the A, B, C, and D are eight bit numbers, wherein the steps of generating the approximate filtered sample, generating the correction, masking the correction and generating the filtered sample include executing the instructions:

```
pxor     A_REG, B_REG ;
pxor     C_REG, D_REG ;
por      A_REG, C_REG ;
pavgb    M_REG, B_REG ;
pavgb    N_REG, D_REG ;
movq     B_REG, M_REG ;
pxor     B_REG, N_REG ;
pand     B_REG, CONST ;
pavgb    M_REG, N_REG ;
psubb    M_REG, B_REG ;
```

wherein A_REG is a register that initially includes A, B_REG is a register that initially includes B, C_REG is a register that initially includes C, D_REG is a register that includes D, CONST is a constant that includes the eight-bit integer value one, M_REG is a register that initially includes A, and N_REG is a register that initially includes C.

41. A method in a signal processor for filtering samples in a digital signal, the method comprising:

generating an approximate filtered sample as a function of a first sample, A, a second sample, B, a third sample, C, and a fourth sample, D, wherein A, B, C, and D are fixed-point numbers;

generating a correction as $(A\oplus B)\mathrm{OR}(C\oplus D)\mathrm{OR}(M\oplus N)$, wherein $M=A+B+1>>1$ and $N=C+D+1>>1$;

masking the correction; and

generating a filtered sample by modifying the approximate filtered sample with the correction.

42. A method in a signal processor for filtering samples in a digital signal, the method comprising:

generating $M1 = (A+B+1)>>1$, wherein A and B are samples in the digital signal;

generating $M2 = (C+D+1)>>1$, wherein C and D are samples in the digital signal;

generating $M3 = (E+F+1)>>1$, wherein E and F are samples in the digital signal;

generating $M4 = (G+H+1)>>1$, wherein G and H are samples in the digital signal;

generating $N1 = (M1+(M3>>1)+(M3>>2))>>2$;

generating $N2= (M2+(M2>>1)+(M4>>2))>>2$; and

generating a filtered sample as $M1+N1-N2$.

43. A computer program product comprising:

a computer readable storage medium having computer program code embodied therein for quantizing a digital signal, the computer program code comprising:

code for generating an approximate filtered sample as a function of less than four samples of the digital signal;

code for generating a correction as a function of the less than four samples; and

code for generating a filtered sample by modifying the approximate filtered sample with the correction.

44. A system for filtering samples in a digital signal, the system comprising:

a memory that stores samples in the digital signal; and

a processor coupled to the memory and operable to perform the steps of:

A) generating an approximate filtered sample as a function of less than four samples of the digital signal;

B) generating a correction as a function of the less than four samples; and

C) generating a filtered sample by modifying the approximate filtered sample with the correction.

45. A computer program product comprising:

a computer readable storage medium having computer program code embodied therein for quantizing a digital signal, the computer program code comprising:

code for generating an approximate filtered sample as a function of a first sample, A, a second sample, B, a third sample, C, and a fourth sample, D, wherein A, B, C, and D are fixed-point numbers;

code for generating a correction as $((A \oplus B)\mathrm{AND}(C \oplus D))\mathrm{OR}(M \oplus N)$, wherein $M=A+B+1>>1$ and $N=C+D+1>>1$;

code for masking the correction; and

code for generating a filtered sample by modifying the approximate filtered sample with the correction.

46. A system for filtering samples in a digital signal, the system comprising:

a memory that stores samples in the digital signal; and

a processor coupled to the memory and operable to perform the steps of:

A) generating an approximate filtered sample as a function of a first sample, A, a second sample, B, a third sample, C, and a fourth sample, D, wherein A, B, C, and D are fixed-point numbers;

B) generating a correction as $((A \oplus B)\mathrm{AND}(C \oplus D))\mathrm{OR}(M \oplus N)$, wherein $M=A+B+1>>1$ and $N=C+D+1>>1$;

C) masking the correction; and

D) generating a filtered sample by modifying the approximate filtered sample with the correction.